Terms used **undo redo**

Found **554** of **193,448**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A temporal model for multi-level undo and redo

W. Keith Edwards, Takeo Igarashi, Anthony LaMarca, Elizabeth D. Mynatt

November 2000 **Proceedings of the 13th annual ACM symposium on User interface software and technology**

Publisher: ACM Press

Full text available: pdf(264.83 KB) Additional Information: full citation, references, citings, index terms

Keywords: Flatland, Timewarp, history management, redo, timelines, undo

2 Design: Dynamic hierarchical undo facility in a fine-grained component environment

Hironori Washizaki, Yoshiaki Fukazawa

February 2002 **Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications CRPIT '02**

Publisher: Australian Computer Society, Inc.

Full text available: pdf(838.08 KB) Additional Information: full citation, abstract, references, index terms

The undo facility is essential for interactive application systems. In conventional object-oriented software development, undo facilities have been implemented based on undo frameworks. However, the use of undo frameworks costs a great deal in both the development and maintenance stages. In this paper, we propose a new technique by which an undo facility can easily be implemented in component-based applications using changes of the component properties. However, since the granularity of the comm ...

Keywords: component-based development, undo mechanism, user interfaces

3 US&R: A new framework for redoing (Extended Abstract)

Jeffrey Scott Vitter

April 1984 **ACM SIGPLAN Notices , ACM SIGSOFT Software Engineering Notes , Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments SDE 1**, Volume 19 , 9 Issue 5 , 3

Publisher: ACM Press

Full text available: pdf(916.36 KB) Additional Information: full citation, abstract, references, citings, index terms

US&R (which stands for Undo, Skip, & Redo) is a new interactive approach to user recovery that offers significant advantages over current Undo/Redo packages. In the US&R package, a SKIP or REDO command may be ambiguous, in which case US&R enumerates the logical interpretations of the command and prompts the user both

textually and graphically for the desired choice. US&R also allows new commands to be executed during the redo process. With US&R, novi ...

4 Session 5: Undoing any operation in collaborative graphics editing systems

David Chen, Chengzheng Sun
September 2001 **Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work**
Publisher: ACM Press

Full text available: pdf(249.64 KB) Additional Information: full citation, abstract, references, citings, index terms

Undo is a useful and widely supported feature which can be used to recover from erroneous operations, learn new system features, and explore alternative solutions. The ability to undo any operation at any time is especially important for collaborative editing systems because it can be used to support local or global undo and also multiple undo models. The Any Undo solution presented in this paper is able to undo any operation in collaborative graphics editing systems. The major challenge in desi ...

Keywords: collaborative editing, concurrency control, consistency maintenance, distributed computing, graphics editing, multi-versioning, undo/redo

5 A selective undo mechanism for graphical user interfaces based on command objects

Thomas Berlage
September 1994 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 1 Issue 3
Publisher: ACM Press

Full text available: pdf(1.78 MB) Additional Information: full citation, abstract, references, citings, index terms, review

It is important to provide a recovery operation for applications with a graphical user interface. A restricted linear undo mechanism can conveniently be implemented using object-oriented techniques. Although linear undo provides an arbitrarily long history, it is not possible to undo isolated commands from the history without undoing all following commands. Various undo models have been proposed to overcome this limitation, but they all ignore the problem that in graphical user interfaces a ...

Keywords: command objects, groupware, undo

6 Undo for anyone, anywhere, anytime

James O'Brien, Marc Shapiro
September 2004 **Proceedings of the 11th workshop on ACM SIGOPS European workshop: beyond the PC EW11**
Publisher: ACM Press

Full text available: pdf(71.09 KB) Additional Information: full citation, abstract, references

Computer systems are complex and unforgiving. Users need environments more tolerant of errors, allowing them to correct mistakes and explore alternatives. This is the aim of Joyce. Joyce records application usage across the system in such a way that the semantic relationships between individual operations are preserved. Using this information Joyce enables an exploratory model of undo/redo; the user can navigate, visualize, edit and experiment with the history of the system safe in the knowledge ...

7 A framework for undoing actions in collaborative systems

Atul Prakash, Michael J. Knister
December 1994 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 1 Issue 4
Publisher: ACM Press

Full text available: Additional Information: full citation, abstract, references, citings, index

pdf(2.54 MB) terms

The ability to undo operations is a standard feature in most single-user interactive applications. We propose a general framework for implementing undo in collaborative systems. The framework allows users to reverse their own changes individually, taking into account the possibility of conflicts between different users' operations that may prevent an undo. The proposed framework has been incorporated into DistEdit, a toolkit for building group text editors. Based on our experience with Dist ...

Keywords: DistEdit, computer-supported cooperative work, concurrency control, groupware, selective undo, state recovery, undo, user recovery

8 Concepts and implications of undo for interactive recovery

Robert F. Gordon, George B. Leeman, Clayton H. Lewis

October 1985 **Proceedings of the 1985 ACM annual conference on The range of computing : mid-80's perspective: mid-80's perspective**

Publisher: ACM Press

Full text available: pdf(648.81 KB) Additional Information: full citation, references, citings, index terms

9 A formal approach to undo operations in programming languages

George B. Leeman

January 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 1

Publisher: ACM Press

Full text available: pdf(2.74 MB) Additional Information: full citation, abstract, references, citings, index terms

A framework is presented for adding a general Undo facility to programming languages. A discussion of relevant literature is provided to show that the idea of Undoing pervades several areas in computer science, and even other disciplines. A simple model of computation is introduced, and it is augmented with a minimal amount of additional structure needed for recovery and reversal. Two different interpretations of Undo are motivated with examples. Then, four primitives are defined in a langu ...

10 Undoing actions in collaborative work

Atul Prakash, Michael J. Knister

December 1992 **Proceedings of the 1992 ACM conference on Computer-supported cooperative work**

Publisher: ACM Press

Full text available: pdf(846.89 KB) Additional Information: full citation, references, citings, index terms

Keywords: collaboration, conflict analysis, groupware, undo

11 Undo any operation at any time in group editors

Chengzheng Sun

December 2000 **Proceedings of the 2000 ACM conference on Computer supported cooperative work**

Publisher: ACM Press

Full text available: pdf(183.08 KB) Additional Information: full citation, abstract, references, citings, index terms

The ability to undo operations is an indispensable feature of real-time group editors, but supporting group undo is a difficult problem. None of the existing solutions for group undo is able to support undoing any operation at any time with guaranteed success. In this paper, we contribute a novel group undo solution with such a capability. The basic idea is to interpret an undo command as a concurrent inverse operation by means of operational

transformation, so that an operation is always ...

Keywords: distributed computing, group editors, group undo, groupware, operational transformation

12 Undo as concurrent inverse in group editors

Chengzheng Sun

December 2002 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 9 Issue 4

Publisher: ACM Press

Full text available: pdf(814.03 KB) Additional Information: full citation, abstract, references, citings, index terms

As an important mechanism for error recovery and exploration of alternatives in interactive and collaborative applications, an undo facility should have the capability of undoing any operation at any time. However, supporting undo in collaborative applications is technically challenging and none of the existing group undo solutions is able to offer such a capability. In this article, we contribute an undo solution with such a capability for group text editors. The basic idea is to interpret an u ...

Keywords: Group undo, REDUCE, collaborative applications, computer-supported cooperative work, concurrency control, consistence maintenance, distributed systems, operational transformation

13 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging

C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz

March 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 1

Publisher: ACM Press

Full text available: pdf(5.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

Keywords: buffer management, latching, locking, space management, write-ahead logging

14 Concurrency control and recovery for balanced B-link trees

Ibrahim Jaluta, Seppo Sippu, Eljas Soisalon-Soininen

April 2005 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 14 Issue 2

Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(302.02 KB) Additional Information: full citation, abstract

In this paper we present new concurrent and recoverable B-link-tree algorithms. Unlike previous algorithms, ours maintain the balance of the B-link tree at all times, so that a logarithmic time bound for a search or an update operation is guaranteed under arbitrary sequences of record insertions and deletions. A database transaction can contain any number of operations of the form "fetch the first (or next) matching record", "insert a record", or "delete a reco ...

Keywords: Concurrency control, Recovery, Transaction, Tree-structure modifications

15 A framework for shared applications with a replicated architecture

Thomas Berlage, Andreas Genau

December 1993 **Proceedings of the 6th annual ACM symposium on User interface software and technology**

Publisher: ACM Press

Full text available: pdf(984.32 KB) Additional Information: full citation, references, citings, index terms

Keywords: application framework, command objects, computer-supported cooperative work, history tree, selective undo and redo, user interface management system

16 Research papers: streams: Fault-tolerance in the Borealis distributed stream processing system

Magdalena Balazinska, Hari Balakrishnan, Samuel Madden, Michael Stonebraker

June 2005 **Proceedings of the 2005 ACM SIGMOD international conference on Management of data**

Publisher: ACM Press

Full text available: pdf(612.50 KB) Additional Information: full citation, abstract, references

We present a replication-based approach to fault-tolerant distributed stream processing in the face of node failures, network failures, and network partitions. Our approach aims to reduce the degree of inconsistency in the system while guaranteeing that available inputs capable of being processed are processed within a specified time threshold. This threshold allows a user to trade availability for consistency: a larger time threshold decreases availability but limits inconsistency, while a smal ...

17 B-tree concurrency control and recovery in page-server database systems

Ibrahim Jaluta, Seppo Sippu, Eljas Soisalon-Soininen

March 2006 **ACM Transactions on Database Systems (TODS)**, Volume 31 Issue 1

Publisher: ACM Press

Full text available: pdf(401.86 KB) Additional Information: full citation, abstract, references, index terms

We develop new algorithms for the management of transactions in a page-shipping client-server database system in which the physical database is organized as a sparse B-tree index. Our starvation-free fine-grained locking protocol combines adaptive callbacks with key-range locking and guarantees repeatable-read-level isolation (i.e., serializability) for transactions containing any number of record insertions, record deletions, and key-range scans. Partial and total rollbacks of client transactio ...

Keywords: ARIES, ARIES/CSA, B-tree, cache consistency, callback locking, client-server database system, data shipping, key-range locking, page server, partial rollback, physiological logging, sparse B-tree, structure modification

18 Atomic incremental garbage collection and recovery for a large stable heap

Elliot K. Kolodner, William E. Weihl

June 1993 **ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data SIGMOD '93**, Volume 22 Issue 2

Publisher: ACM Press

Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

A stable heap is storage that is managed automatically using garbage collection, manipulated using atomic transactions, and accessed using a uniform storage model. These features enhance reliability and simplify programming by preventing errors due to explicit deallocation, by masking failures and concurrency using transactions, and by eliminating the distinction between accessing temporary storage and permanent storage. Stable heap management is useful for programming lang ...

19 Model and verification of a data manager based on ARIES

Dean Kuo

December 1996 **ACM Transactions on Database Systems (TODS)**, Volume 21 Issue 4

Publisher: ACM Press

Full text available: pdf(813.93 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In this article, we model and verify a data manager whose algorithm is based on ARIES. The work uses the I/O automata method as the formal model and the definition of correctness is defined on the interface between the scheduler and the data manager.

Keywords: ARIES, I/O automata, system failures

20 Group editing algorithms: Achieving undo in bitmap-based collaborative graphics editing systems

Xueyi Wang, Jiajun Bu, Chun Chen

November 2002 **Proceedings of the 2002 ACM conference on Computer supported cooperative work**

Publisher: ACM Press

Full text available: pdf(239.51 KB) Additional Information: full citation, abstract, references, index terms

Bitmap-based collaborative graphics editing systems are a special class of real-time collaborative editing systems. Undo is an important and difficult problem in these systems. Existing solutions show low efficiency because additional space cost should be added to achieve the function of undo. In this paper, we propose a new solution to resolve the undo problem. The basic idea is to reduce space cost through exploring relations among operations. The algorithm given in the paper can undo any oper ...

Keywords: CSCW, bitmap-based collaborative graphics editing systems, undo

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used **undo routine serial**

Found **610** of **193,448**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A portable syntactic error recovery scheme for LR(1) parsers

Pyda Srisuresh, Michael J. Eager

March 1985 **Proceedings of the 1985 ACM thirteenth annual conference on Computer Science**

Publisher: ACM Press

Full text available: pdf(980.89 KB) Additional Information: full citation, abstract, references, index terms

A 4-level language independent error recovery scheme for table driven LR(1) parsers is presented. The first two levels are intended for recovery with appropriate corrections and the next two for simple recovery without corrections. The objective is to do the recovery without affecting the semantics or data structure of the compiler while at the same time producing necessary diagnostics and terminating gracefully. The scheme is a significant improvement and a step forward in the direction of ...

Keywords: LR(1), compilers, error correction, error recovery, grammars, parsar generators, portability, semantics, syntax

2 Condition handling in SQL persistent stored modules

Jeff Richey

September 1995 **ACM SIGMOD Record**, Volume 24 Issue 3

Publisher: ACM Press

Full text available: pdf(551.98 KB) Additional Information: full citation, abstract, index terms

The national and international standards committees responsible for Database Language SQL have proposed a candidate extension for SQL Persistent Stored Modules (SQL/PSM). The purpose of this extension is to provide a computationally complete language for the declaration and invocation of SQL stored modules and routines. Typically, such routines are stored in a database Server and executed from an application Client in a Client/Server environment.The proposed SQL/PSM consists of syntax and semant ...

3 Courses: An introduction to sketch-based interfaces

Joseph LaViola, Randall Davis, Takeo Igarashi

July 2006 **Material presented at the ACM SIGGRAPH 2006 conference SIGGRAPH '06**

Publisher: ACM Press

Full text available: pdf(31.58 MB) Additional Information: full citation, abstract

Sketch-based interfaces are a natural, pencil-and-paper-like approach to interacting with a variety of applications, including conceptual modeling, animation, and note-taking systems. This course offers an in-depth discussion of sketch-based interface design, ranging from simple gestural commands to complex sketch-understanding systems. Attendees will learn how these interfaces are designed and how to develop their own.

4 Linguistic support for atomic data types

William E. Weihl

April 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 2

Publisher: ACM Press

Full text available: pdf(2.10 MB) Additional Information: full citation, abstract, references, citings, index terms

The problems of concurrency and failures in distributed systems can be addressed by implementing applications in terms of atomic data types: data types whose objects provide serializability and recoverability for transactions using them. The specifications of the types can be used to permit high levels of concurrency among transactions while still ensuring atomicity. However, highly concurrent implementations can be quite complicated. In this paper we analyze the expressive power of existin ...

5 Efficient algorithms for bidirectional debugging

Bob Boothe

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35 Issue 5

Publisher: ACM Press

Full text available: pdf(473.76 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses our research into algorithms for creating an efficient bidirectional debugger in which all traditional forward movement commands can be performed with equal ease in the reverse direction. We expect that adding these backwards movement capabilities to a debugger will greatly increase its efficacy as a programming tool. The efficiency of our methods arises from our use of event counters that are embedded into the program being debugged. These counters are used ...

6 The most important technical library in the world

Bruce Greer

December 1998 **ACM SIGPLAN Fortran Forum**, Volume 17 Issue 3

Publisher: ACM Press

Full text available: pdf(603.84 KB) Additional Information: full citation, abstract, index terms

Intel provides several libraries known as the Performance Library Suite to address several computational areas: signal processing, image processing, speech and character recognition and linear algebra. This latter library is known as the Math Kernel Library, or MKL.In this paper we discuss the methods we use to achieve high performance on the current version. These methods revolve primarily around the management of memory and minimization of branch costs. Memory management involves compensating ...

7 Interactive Editing Systems: Part I

Norman Meyrowitz, Andries van Dam

September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Publisher: ACM Press

Full text available: pdf(3.08 MB) Additional Information: full citation, citings, index terms

8 An integrating, transformation-oriented approach to concurrency control and undo in group editors

Matthias Ressel, Doris Nitsche-Ruhland, Rul Gunzenhäuser

November 1996 **Proceedings of the 1996 ACM conference on Computer supported cooperative work**

Publisher: ACM Press

Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

Keywords: concurrency control, group editors, group undo, groupware, interaction model, operation transformation

9 On reversible subroutines and computers that run backwards

E. D. Reilly, F. D. Federighi
September 1965 **Communications of the ACM**, Volume 8 Issue 9
Publisher: ACM Press
Full text available: pdf(478.37 KB) Additional Information: full citation, abstract, citings, index terms

A computer design is described which permits subroutines to be executed backward as well as forward, either with their instructions unchanged or replaced with conjugate instructions. It is shown that using this concept a number of new subroutine types can be developed with rather unusual properties. Since these properties are analogous to certain matrix operations, a arallel nomenclature is suggested for their classification.

10 Locking Expressions for Increased Database Concurrency

Anthony Klug
January 1983 **Journal of the ACM (JACM)**, Volume 30 Issue 1
Publisher: ACM Press
Full text available: pdf(1.16 MB) Additional Information: full citation, references, citings, index terms

11 Transporting a portable operating system: UNIX to an IBM minicomputer

Paul J. Jalics, Thomas S. Heines
December 1983 **Communications of the ACM**, Volume 26 Issue 12
Publisher: ACM Press
Full text available: pdf(772.83 KB) Additional Information: full citation, abstract, references, citings, index terms

The "portable" UNIX operating system was transported to an IBM Series/1 minicomputer. The process of transporting is described with emphasis on (1) adapting to the target machine architecture; (2) the selection of the approach taken to transporting; (3) a description of the problems encountered; (4) the degree of portability of the UNIX system; and (5) a summary of the portability lessons learned.

Keywords: C, UNIX, operating system, transporting

12 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging

C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz
March 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 1
Publisher: ACM Press
Full text available: pdf(5.23 MB) Additional Information: full citation, abstract, references, citings, index terms, review

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

Keywords: buffer management, latching, locking, space management, write-ahead logging

13 Algol 68 as an implementation language for portable interpreters

Frank G. Pagan
March 1977 **ACM SIGPLAN Notices , Proceedings of the Strathclyde ALGOL 68 conference**, Volume 12 Issue 6
Publisher: ACM Press
Full text available: pdf(553.72 KB) Additional Information: full citation, abstract, references, index terms

By making use of its advanced and highly expressive facilities, Algol 68 can be used to implement interpretive language processors with an unusual degree of conceptual clarity and machine independence. The internal representations of source programs in such a processor consist of high-level data structures which are interpreted by means of a set of readable, mutually recursive Algol 68 procedures. The technique is illustrated by applying it to the implementation of a miniature sample langua ...

14 Commutativity analysis: a new analysis technique for parallelizing compilers

Pedro C. Diniz
November 1997 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 19 Issue 6
Publisher: ACM Press
Full text available: pdf(472.62 KB) Additional Information: full citation, abstract, references, citings, index terms

This article presents a new analysis technique, commutativity analysis, for automatically parallelizing computations that manipulate dynamic, pointer-based data structures. Commutativity analysis views the computation as composed of operations on objects. It then analyzes the program at this granularity to discover when operations commute (i.e., generate the same final result regardless of the order in which they execute). If all of the operations required to perform a given computation com ...

Keywords: parallel computing

15 Problems in supporting data base transactions in an operating system transaction manager

Michael Stonebraker, Deborah DuBourdieux, William Edwards
January 1985 **ACM SIGOPS Operating Systems Review**, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(491.85 KB) Additional Information: full citation, abstract, references

This paper reports on the experience of the authors in attempting to support data base transactions on top of an existing operating system transaction manager. It will be seen that significant modifications to both the example data base system and the example operating system are required to support the concept. The conclusion to be drawn is that operating system transaction managers will have to be designed more generally than is now suggested and that application programs (such as data base sy ...

16 Courses: State of the art in interactive ray tracing

Peter Shirley
July 2006 **Material presented at the ACM SIGGRAPH 2006 conference SIGGRAPH '06**
Publisher: ACM Press
Full text available: pdf(14.08 MB) Additional Information: full citation, abstract

Recent improvements in computer hardware have allowed ray tracing to be used in some interactive applications. The trends in architecture and expansions of geometric model should increase the use of interactive ray tracing. This course presents recent and often not-yet published work on interactive ray tracing.

17 A formal approach to undo operations in programming languages

George B. Leeman

January 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 1
Publisher: ACM Press

Full text available: pdf(2.74 MB) Additional Information: full citation, abstract, references, citings, index terms

A framework is presented for adding a general Undo facility to programming languages. A discussion of relevant literature is provided to show that the idea of Undoing pervades several areas in computer science, and even other disciplines. A simple model of computation is introduced, and it is augmented with a minimal amount of additional structure needed for recovery and reversal. Two different interpretations of Undo are motivated with examples. Then, four primitives are defined in a langu ...

18 A serialization graph construction for nested transactions

Alan Fekete, Nancy Lynch, William E. Weihl
April 1990 **Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**
Publisher: ACM Press

Full text available: pdf(2.23 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper makes three contributions. First, we present a proof technique that offers system designers the same ease of reasoning about nested transaction systems as is given by the classical theory for systems without nesting, and yet can be used to verify that a system satisfies the robust "user view" definition of correctness of [10]. Second, as applications of the technique, we verify the correctness of Moss' read/write locking algorithm for nested transactions, and of an un ...

19 An APL system for experimentation with spectral methods in fault detection

Eric Gullichsen
September 1986 **ACM SIGAPL APL Quote Quad**, Volume 17 Issue 1
Publisher: ACM Press

Full text available: pdf(563.15 KB) Additional Information: full citation, abstract, index terms

An ability to detect faults in digital circuitry is of increasing importance as a result of the current trend towards ever-greater levels of circuit complexity in VLSI design. This paper describes a system of APL routines that permits the interactive specification of combinational circuits and the discovery of spectra for lines of the (fault-free) circuits. The system can also simulate the spectral behavior of a specified circuit under the condition of "stuck-at" faults on any combination of cir ...

20 Interactive Editing Systems: Part II

Norman Meyrowitz, Andries van Dam
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press

Full text available: pdf(9.17 MB) Additional Information: full citation, references, citings, index terms

Terms used **add undo**

Found **1,618** of **193,448**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Design: Dynamic hierarchical undo facility in a fine-grained component environment

Hironori Washizaki, Yoshiaki Fukazawa

February 2002 **Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications CRPIT '02**

Publisher: Australian Computer Society, Inc.

Full text available: pdf(838.08 KB) Additional Information: full citation, abstract, references, index terms

The undo facility is essential for interactive application systems. In conventional object-oriented software development, undo facilities have been implemented based on undo frameworks. However, the use of undo frameworks costs a great deal in both the development and maintenance stages. In this paper, we propose a new technique by which an undo facility can easily be implemented in component-based applications using changes of the component properties. However, since the granularity of the comm ...

Keywords: component-based development, undo mechanism, user interfaces

2 A formal approach to undo operations in programming languages

George B. Leeman

January 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 1

Publisher: ACM Press

Full text available: pdf(2.74 MB) Additional Information: full citation, abstract, references, citings, index terms

A framework is presented for adding a general Undo facility to programming languages. A discussion of relevant literature is provided to show that the idea of Undoing pervades several areas in computer science, and even other disciplines. A simple model of computation is introduced, and it is augmented with a minimal amount of additional structure needed for recovery and reversal. Two different interpretations of Undo are motivated with examples. Then, four primitives are defined in a langu ...

3 Dynamic space management for user interfaces

Blaine A. Bell, Steven K. Feiner

November 2000 **Proceedings of the 13th annual ACM symposium on User interface software and technology**

Publisher: ACM Press

Full text available: pdf(586.22 KB) Additional Information: full citation, references, citings, index terms

Keywords: display layout, geometric modeling, overlap avoidance, space allocation, spatial data structures, user interface design, window management

4 An automated tool for analyzing completeness of equational specifications

Deepak Kapur

August 1994 **Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis**

Publisher: ACM Press

Full text available: pdf(1.89 MB) Additional Information: full citation, abstract, references, index terms

Books on software engineering methodologies talk about the significance and need for designing consistent and complete specifications during the requirement analysis and design stages of a software development cycle. There is, however, little (or at best very limited) discussion of methods for ensuring these structural properties of specifications. In this paper, we discuss methods for checking completeness of equational specifications. Some of these methods were earlier proposed in somewha ...

5 Cooperative visual manipulation of music notation

P. Bellini, P. Nesi, M. B. Spinu

September 2002 **ACM Transactions on Computer-Human Interaction (TOCHI)**, Volume 9 Issue 3

Publisher: ACM Press

Full text available: pdf(3.42 MB) Additional Information: full citation, abstract, references, citings, index terms

As computer technologies and their potential emerging applications spread out, new needs have been detected for computer-based applications of music; cooperative music notation editing both in orchestras and music schools is one of them. This article is the only public document describing the details of cooperative work on music notation of MOODS (Music Object Oriented Distributed System). MOODS is a synchronous real-time cooperative editor for music scores. Its architecture includes mechanisms ...

Keywords: Collaboration of music notation editing, additional command list, collaborative systems, computer-supported cooperative work, consistency control, cooperative music, distributed music, electronic lectern, neutral version, selective undo, user interface management systems

6 Group editing algorithms: Achieving undo in bitmap-based collaborative graphics editing systems

Xueyi Wang, Jiajun Bu, Chun Chen

November 2002 **Proceedings of the 2002 ACM conference on Computer supported cooperative work**

Publisher: ACM Press

Full text available: pdf(239.51 KB) Additional Information: full citation, abstract, references, index terms

Bitmap-based collaborative graphics editing systems are a special class of real-time collaborative editing systems. Undo is an important and difficult problem in these systems. Existing solutions show low efficiency because additional space cost should be added to achieve the function of undo. In this paper, we propose a new solution to resolve the undo problem. The basic idea is to reduce space cost through exploring relations among operations. The algorithm given in the paper can undo any oper ...

Keywords: CSCW, bitmap-based collaborative graphics editing systems, undo

7 Declarative event-oriented programming

Conal Elliott

September 2000 **Proceedings of the 2nd ACM SIGPLAN international conference on Principles and practice of declarative programming**

Publisher: ACM Press

Full text available: pdf(1.26 MB) Additional Information: full citation, references, citings, index terms

8 Understanding programs and interfaces: Answering why and why not questions in user interfaces

Brad A. Myers, David A. Weitzman, Andrew J. Ko, Duen H. Chau

April 2006 **Proceedings of the SIGCHI conference on Human Factors in computing systems CHI '06**

Publisher: ACM Press

Full text available: pdf(624.29 KB) Additional Information: full citation, abstract, references, index terms

Modern applications such as Microsoft Word have many automatic features and hidden dependencies that are frequently helpful but can be mysterious to both novice and expert users. The ""Crystal"" application framework provides an architecture and interaction techniques that allow programmers to create applications that let the user ask a wide variety of questions about why things did and did not happen, and how to use the related features of the application without using natural language. A user ...

Keywords: help, natural programming, questions, why

9 Simplify: a theorem prover for program checking

David Detlefs, Greg Nelson, James B. Saxe

May 2005 **Journal of the ACM (JACM)**, Volume 52 Issue 3

Publisher: ACM Press

Full text available: pdf(1.93 MB) Additional Information: full citation, abstract, references, index terms

This article provides a detailed description of the automatic theorem prover Simplify, which is the proof engine of the Extended Static Checkers ESC/Java and ESC/Modula-3. Simplify uses the Nelson--Oppen method to combine decision procedures for several important theories, and also employs a matcher to reason about quantifiers. Instead of conventional matching in a term DAG, Simplify matches up to equivalence in an E-graph, which detects many relevant pattern instances that would be missed by th ...

Keywords: Theorem proving, decision procedures, program checking

10 Supporting nested transactional memory in logTM

Michelle J. Moravan, Jayaram Bobba, Kevin E. Moore, Luke Yen, Mark D. Hill, Ben Liblit, Michael M. Swift, David A. Wood

October 2006 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , Proceedings of the 12th international conference on Architectural support for programming languages and operating systems ASPLOS-XII**, Volume 41 , 40 , 34 Issue 11 , 5 , 5

Publisher: ACM Press

Full text available: pdf(239.03 KB) Additional Information: full citation, abstract, references, index terms

Nested transactional memory (TM) facilitates software composition by letting one module invoke another without either knowing whether the other uses transactions. **Closed nested transactions** extend isolation of an inner transaction until the toplevel transaction commits. Implementations may flatten nested transactions into the top-level one, resulting in a complete abort on conflict, or allow partial abort of inner transactions. **Open nested transactions** allow a committing inner tran ...

Keywords: logTM, nesting, transactional memory

11 Anatomy of a native XML base management system

T. Fiebig, S. Helmer, C.-C. Kanne, G. Moerkotte, J. Neumann, R. Schiele, T. Westmann

December 2002 **The VLDB Journal — The International Journal on Very Large Data**

Bases, Volume 11 Issue 4
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(300.97 KB) Additional Information: full citation, abstract, citings, index terms

Several alternatives to manage large XML document collections exist, ranging from file systems over relational or other database systems to specifically tailored XML base management systems. In this paper we give a tour of Natix, a database management system designed from scratch for storing and processing XML data. Contrary to the common belief that management of XML data is just another application for traditional databases like relational systems, we illustrate how almost every component in a ...

Keywords: Database, XML

12 Concurrency and recovery in generalized search trees

Marcel Kornacker, C. Mohan, Joseph M. Hellerstein
June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data SIGMOD '97**, Volume 26 Issue 2
Publisher: ACM Press
Full text available: pdf(1.59 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

13 VODAK open nested transactions—visualizing database internals

Peter Muth, Thomas C. Rakow
June 1993 **ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data SIGMOD '93**, Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(207.07 KB) Additional Information: full citation, abstract, references, citings, index terms

VODAK is a prototype of an object-oriented, distributed database system developed during the past five years at the Integrated Publication and Information Systems Institute (IPSI). The aim of demonstrating VODAK Open Nested Transactions is to provide insights into internals of database systems that are usually hidden from application programmers and users. By utilizing semantics of methods, VODAK Open Nested Transactions increase the degree of parallelism between concurrent transactions com ...

14 Scalable parallel allocation: McRT-Malloc: a scalable transactional memory allocator

Richard L. Hudson, Bratin Saha, Ali-Reza Adl-Tabatabai, Benjamin C. Hertzberg
June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**
Publisher: ACM Press
Full text available: pdf(332.45 KB) Additional Information: full citation, abstract, references, index terms

Emerging multi-core processors promise to provide an exponentially increasing number of hardware threads with every generation. Applications will need to be highly concurrent to fullyuse the power of these processors. To enable maximum concurrency, libraries (such as malloc-free packages) would therefore need to use non-blocking algorithms. But lock-free algorithms are notoriously difficult to reason about and inappropriate for average programmers. Transactional memory promises to significantly ...

Keywords: memory management, runtimes, synchronization, transactional memory

15

Interface tools: Interacting with a visual editor
Roberta Mancini
May 1996 **Proceedings of the workshop on Advanced visual interfaces**
Publisher: ACM Press
Full text available: pdf(1.04 MB) Additional Information: full citation, abstract, references

In this paper, we investigate the problem of querying a database of images. In order to improve the communication between human and computer, we propose a visual editor as an interaction tool. Really, the most simple way to formulate a query to a database of images is to allow the user to draw a sketch of the picture he is interested in. This sketch will be used to formulate a query within the visual query system. This editor, called VisEd, has been developed following a formal model (the PIE mo ...

Keywords: formal model, reachability, undo, visual editor, visual query system

16 Interactive Editing Systems: Part II
Norman Meyrowitz, Andries van Dam
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(9.17 MB) Additional Information: full citation, references, citings, index terms

17 Coding between mouse and keyboard, part I
Patricia Jung
September 2002 **Linux Journal**, Volume 2002 Issue 101
Publisher: Specialized Systems Consultants, Inc.
Full text available: html(19.65 KB) Additional Information: full citation, abstract, index terms

In the first part of this two-part article, Jung provides a working example of building GUI apps with Qt.

18 Theoretical foundations for compensations in flow composition languages
Roberto Bruni, Hernán Melgratti, Ugo Montanari
January 2005 **ACM SIGPLAN Notices , Proceedings of the 32nd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '05**, Volume 40 Issue 1
Publisher: ACM Press
Full text available: pdf(288.25 KB) Additional Information: full citation, abstract, references, citings, index terms

A key aspect when aggregating business processes and web services is to assure transactional properties of process executions. Since transactions in this context may require long periods of time to complete, traditional mechanisms for guaranteeing atomicity are not always appropriate. Generally the concept of long running transactions relies on a weaker notion of atomicity based on compensations. For this reason, programming languages for service composition cannot leave out two key aspects: ...

Keywords: *compensations, process description languages, transactions*

19 Session 5: Undoing any operation in collaborative graphics editing systems
David Chen, Chengzheng Sun
September 2001 **Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work**
Publisher: ACM Press
Full text available: pdf(249.64 KB) Additional Information: full citation, abstract, references, citings, index terms

Undo is a useful and widely supported feature which can be used to recover from

erroneous operations, learn new system features, and explore alternative solutions. The ability to undo any operation at any time is especially important for collaborative editing systems because it can be used to support local or global undo and also multiple undo models. The Any Undo solution presented in this paper is able to undo any operation in collaborative graphics editing systems. The major challenge in desi ...

Keywords: collaborative editing, concurrency control, consistency maintenance, distributed computing, graphics editing, multi-versioning, undo/redo

20 Optimizing memory transactions

Tim Harris, Mark Plesko, Avraham Shinnar, David Tarditi

June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6

Publisher: ACM Press

Full text available: pdf(235.79 KB) Additional Information: full citation, abstract, references, index terms

Atomic blocks allow programmers to delimit sections of code as 'atomic', leaving the language's implementation to enforce atomicity. Existing work has shown how to implement atomic blocks over *word-based transactional memory* that provides scalable multi-processor performance without requiring changes to the basic structure of objects in the heap. However, these implementations perform poorly because they interpose on all accesses to shared memory in the atomic block, redirecting updates t ...

Keywords: atomicity, critical regions, transactional memory

Results for "((undo redo)<in>metadata)"

e-mail printer friendly

Your search matched **1** of **1450046** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

((undo redo)<in>metadata) Search

Check to search only within this results set

Display Format: Citation Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

view selected items Select All Deselect All

1. **Experience with evaluating human-assisted recovery processes**
   Brown, A.B.; Chung, L.; Kakes, W.; Ling, C.; Patterson, D.A.;
   Dependable Systems and Networks, 2004 International Conference on
   28 June-1 July 2004 Page(s):405 - 410
   Digital Object Identifier 10.1109/DSN.2004.1311910
   AbstractPlus | Full Text: PDF(299 KB) **IEEE CNF**
   Rights and Permissions